CLAIMS

We claim:

1. A method for reducing total code size, comprising the steps of:

determining a latency between a defining instruction and a using instruction and

inserting a NOP field into at least one of said defining and using instruction.

2. The method of claim 1, wherein said NOP field is inserted into said defining instruction.

3. The method of claim 1, wherein said NOP field is inserted into said using instruction.

4. The method of claim 2, wherein said NOP field is inserted at an end of said defining instruction.

5. The method of claim 3, wherein said NOP field is inserted at an end of said defining instruction.

6. A method for reducing total code size during branching, comprising the steps of

determining a latency after a branch instruction for initiating a branch from a first point to a second point in an instruction stream, and

inserting a NOP field into said branch instruction.

7. The method of claim 6, wherein said branch NOP field is affixed to an end of said branch instruction.

8 An apparatus having reduced total code size, comprising a processor including at least one defining instruction followed by at least one using instruction, wherein a latency exists between said at least one defining instruction and said at least one using instruction and wherein at least one of said at least one defining and using instruction includes a NOP field.

9. The apparatus of claim 8 wherein said NOP field is inserted into said defining instruction.

10. The apparatus of claim 8, wherein said NOP field is inserted into said using instruction.

11. The apparatus of claim 9, wherein said NOP field is inserted at an end of said defining instruction.

12. The apparatus of claim 10, wherein said NOP field is inserted at an end of said defining instruction.

13. An apparatus for reducing total code size during branching, comprising a processor including at least one branch instruction for branching from a first point to a second point in an instruction stream, , such that a latency exists in the branching between said first point and said second point, and said at least one branch instruction includes a NOP field .

14. The apparatus of claim 13, wherein said branch NOP field is affixed to an end of said branch instruction.

15. A method for reducing total code size comprising the steps of locating at least one delayed effect instruction followed by NOPs within a code; deleting said NOPs from said code; and inserting a NOP field into a delaying instruction.

16. The method of claim 15, wherein said delaying instruction is said locating at least one delayed effect instruction.

17. The method of claim 15, wherein a portion of said deleted NOPS precede said delaying instruction.

18. The method of claim 15, wherein said delayed effect instruction is a load instruction.

19. The method of claim 15, wherein said delayed effect instruction is a branch instruction.

20. An apparatus for reducing total code size comprising a processor including a code containing at least one delayed effect instruction, wherein said least one of the at least one delayed effect instructions includes a NOP field.

21. The apparatus of claim 20, wherein said at least one delayed effect instruction is a load instruction.

22. The apparatus of claim 20, wherein said at least one delayed effect instruction is a branch instruction.